Credit Score Classification

Python & Machine Learning Project

İsmail Aksu

Table of Contents:

1.Introduction

Project Objective:

The aim of this project is to develop a classification model for credit score assessment. Credit score is a crucial metric used to evaluate the financial status of individuals or organizations. The primary objective of this project is to leverage data analysis and machine learning techniques to develop a classification model that assesses customers' credit risk.

The development of such a model plays a significant role in the lending process of financial institutions. An accurate credit score classification model can assist financial institutions in evaluating loan applications more accurately and identifying risky customers to mitigate credit risk. This, in turn, enables institutions to conduct more effective risk management and enhances financial stability.

The data analysis and machine learning techniques employed in this project are essential for understanding the complex relationships underlying credit score classification and predicting customers' credit risk. These techniques can identify patterns in the dataset, determine factors influencing credit risk, and select appropriate features to improve the model's accuracy.

In conclusion, this project aims to contribute to the improvement of credit score classification, making it more accurate and reliable. A robust credit score classification model will enhance both the risk management practices of financial institutions and provide customers with a fairer credit assessment process.

2. Data Collection and Preprocessing

During the data analysis phase of my project, various analyzes were performed on the data set obtained from "https://statso.io/credit-score-classification-case-study/".This dataset includes information about customers' incomes, ages, loan amounts, payment histories, and other financial and demographic attributes. Using exploratory data analysis (EDA), we examined the structural characteristics of the dataset and relationships between variables. Additionally, we addressed missing data, standardized numerical variables, and performed feature selection to identify important features. This step laid the foundation for subsequent model development and evaluation processes.

Data: https://statso.io/credit-score-classification-case-study/

3.Variable Analysis (Python):

#First, we import our Python libraries.

```
import pandas as pd
import numpy as np
import plotly.express as px
import plotly.graph_objects as go
import plotly.io as pio
pio.templates.default = "plotly_white"
```

#upload data

```
data = pd.read_excel("train.xlsx")
```

#Let's examine the first 5 of our data.

```
data.head()
```

	ID	Customer_ID	Month	Name	Age	SSN	Occupation	Annual_Income	Monthly_Inhand_Salary	Num_Bank_Accounts	...	Credit_Mix	Outstanding
0	5634	3392	1	Aaron Maashoh	23.0	821000265.0	Scientist	19114.12	1.824843e+16	3.0	...	Good	
1	5635	3392	2	Aaron Maashoh	23.0	821000265.0	Scientist	19114.12	1.824843e+16	3.0	...	Good	
2	5636	3392	3	Aaron Maashoh	23.0	821000265.0	Scientist	19114.12	1.824843e+16	3.0	...	Good	
3	5637	3392	4	Aaron Maashoh	23.0	821000265.0	Scientist	19114.12	1.824843e+16	3.0	...	Good	
4	5638	3392	5	Aaron Maashoh	23.0	821000265.0	Scientist	19114.12	1.824843e+16	3.0	...	Good	

5 rows × 28 columns

#Let’s have a look at the information about the columns in the dataset:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 28 columns):
 #   Column                    Non-Null Count   Dtype  
---  ------                    --------------   -----  
 0   ID                        100000 non-null  int64  
 1   Customer_ID               100000 non-null  int64  
 2   Month                     100000 non-null  int64  
 3   Name                      100000 non-null  object 
 4   Age                       100000 non-null  float64
 5   SSN                       100000 non-null  float64
 6   Occupation                100000 non-null  object 
 7   Annual_Income             100000 non-null  float64
 8   Monthly_Inhand_Salary     100000 non-null  float64
 9   Num_Bank_Accounts         100000 non-null  float64
 10  Num_Credit_Card           100000 non-null  float64
 11  Interest_Rate             100000 non-null  float64
 12  Num_of_Loan               100000 non-null  float64
 13  Type_of_Loan              100000 non-null  object 
 14  Delay_from_due_date       100000 non-null  float64
 15  Num_of_Delayed_Payment    100000 non-null  float64
 16  Changed_Credit_Limit      100000 non-null  object 
 17  Num_Credit_Inquiries      100000 non-null  float64
 18  Credit_Mix                100000 non-null  object 
 19  Outstanding_Debt          100000 non-null  object 
 20  Credit_Utilization_Ratio  100000 non-null  float64
 21  Credit_History_Age        100000 non-null  float64
 22  Payment_of_Min_Amount     100000 non-null  object 
 23  Total_EMI_per_month       100000 non-null  float64
 24  Amount_invested_monthly   100000 non-null  float64
 25  Payment_Behaviour         100000 non-null  object 
 26  Monthly_Balance           100000 non-null  float64
 27  Credit_Score              100000 non-null  object 
```

#Before moving forward, let’s have a look if the dataset has any null values or not:

```
data.isnull().sum()
```

```
ID                          0
Customer_ID                 0
Month                       0
Name                        0
Age                         0
SSN                         0
Occupation                  0
Annual_Income               0
Monthly_Inhand_Salary       0
Num_Bank_Accounts           0
Num_Credit_Card             0
Interest_Rate               0
Num_of_Loan                 0
Type_of_Loan                0
Delay_from_due_date         0
Num_of_Delayed_Payment      0
Changed_Credit_Limit        0
Num_Credit_Inquiries        0
Credit_Mix                  0
Outstanding_Debt            0
Credit_Utilization_Ratio    0
Credit_History_Age          0
Payment_of_Min_Amount       0
Total_EMI_per_month         0
Amount_invested_monthly     0
Payment_Behaviour           0
Monthly_Balance             0
Credit_Score                0
dtype: int64
```

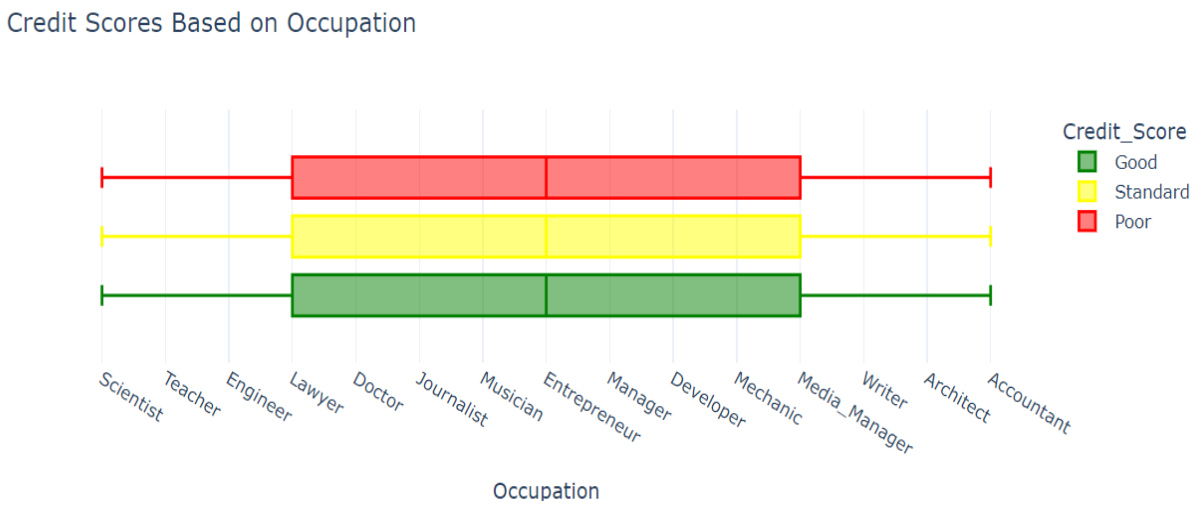
#The dataset doesn't have any null values. As this dataset is labelled, let's have a look at the Credit_Score column values:

```
data["Credit_Score"].value_counts()
```

```
Credit_Score
Standard    53174
Poor        28998
Good        17828
Name: count, dtype: int64
```

#I will start by exploring the occupation feature to know if the occupation of the person affects credit scores:

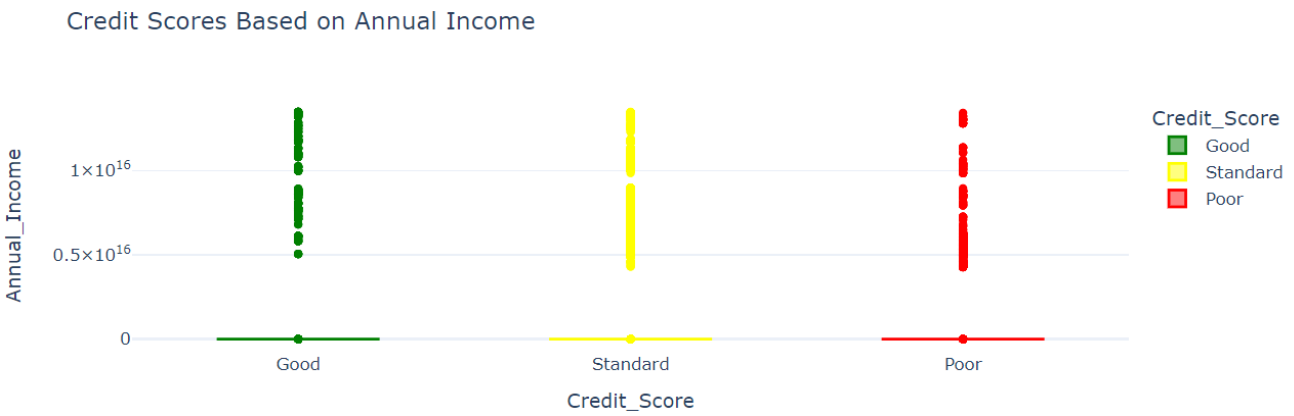
```
fig= px.box(data,
            x="Occupation",
            color = "Credit_Score",
            title ="Credit Scores Based on Occupation",
            color_discrete_map = {"Poor":"red",
                                  "Standard" :"yellow",
                                  "Good" : "green"})
fig.show()
```



#There's not much difference in the credit scores of all occupations mentioned in the data.

#Now let’s explore whether the Annual Income of the person impacts your credit scores or not:

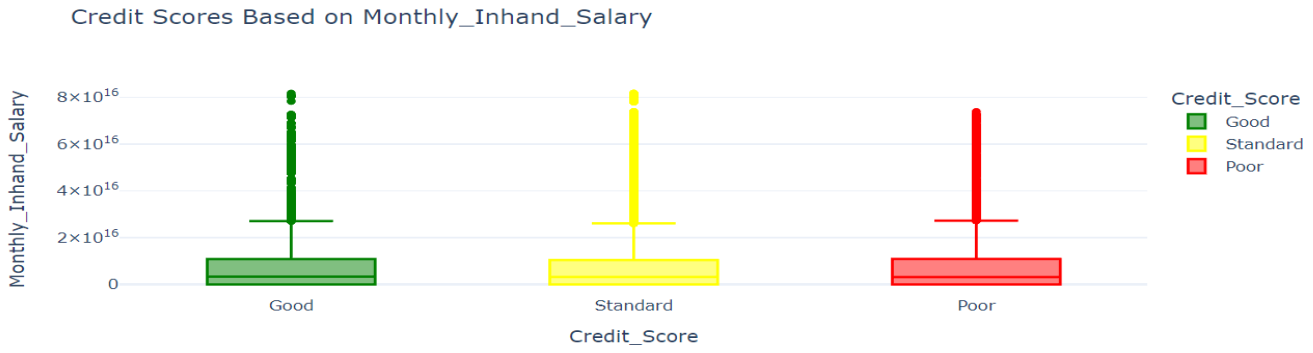
```
fig = px.box(data,
             x = "Credit_Score",
             y = "Annual_Income",
             color = "Credit_Score",
             title ="Credit Scores Based on Annual Income",
             color_discrete_map = {"Poor":"red",
                                   "Standard" :"yellow",
                                   "Good" : "green"})
fig.update_traces(quartilemethod = "exclusive")
fig.show()
```



#According to the above visualization, the more you earn annually, the better your credit score is.

#Now let’s explore whether the monthly in-hand salary impacts credit scores or not:

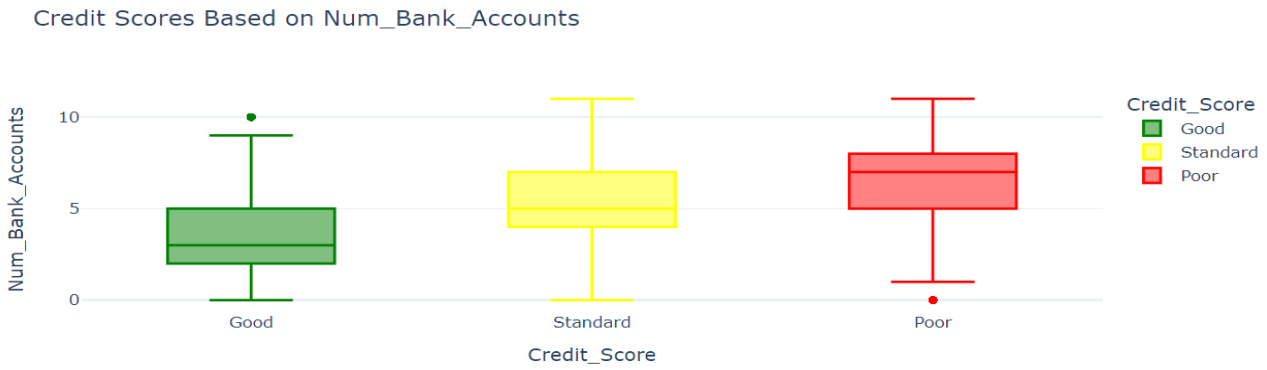
```
fig = px.box(data,
             x = "Credit_Score",
             y = "Monthly_Inhand_Salary",
             color = "Credit_Score",
             title ="Credit Scores Based on Monthly_Inhand_Salary",
             color_discrete_map = {"Poor":"red",
                                   "Standard" :"yellow",
                                   "Good" : "green"})
fig.update_traces(quartilemethod = "exclusive")
fig.show()
```



#like annual income, the more monthly in-hand salary you earn, the better your credit score will become.

#Now let's see if having more bank accounts impacts credit scores or not:

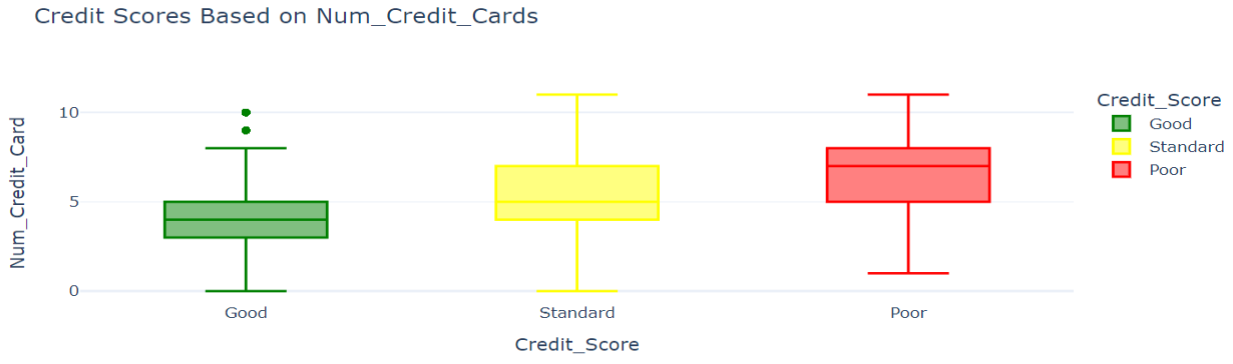
```
fig = px.box(data,
             x = "Credit_Score",
             y = "Num_Bank_Accounts",
             color = "Credit_Score",
             title ="Credit Scores Based on Num_Bank_Accounts",
             color_discrete_map = {"Poor":"red",
                                   "Standard" :"yellow",
                                   "Good" : "green"})
fig.update_traces(quartilemethod = "exclusive")
fig.show()
```



#Maintaining more than five accounts is not good for having a good credit score .A person should have 2 – 3 bank accounts only. So having more bank accounts doesn't positively impact credit scores.

#Now let's see the impact on credit scores based on the number of credit cards you have:

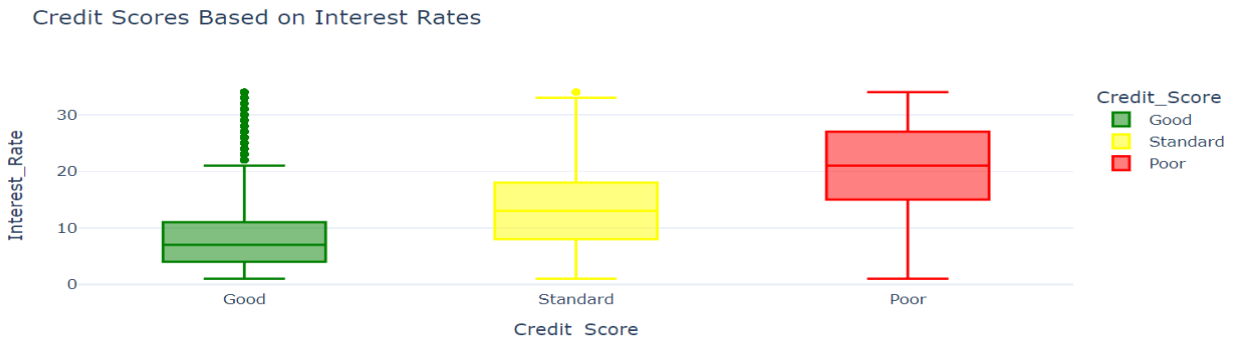
```
fig = px.box(data,
             x = "Credit_Score",
             y = "Num_Credit_Card",
             color = "Credit_Score",
             title ="Credit Scores Based on Num_Credit_Cards",
             color_discrete_map = {"Poor":"red",
                                   "Standard" :"yellow",
                                   "Good" : "green"})
fig.update_traces(quartilemethod = "exclusive")
fig.show()
```



#Justlike bank accounts, having more credit cards will not positively impact your credit scores.Having 3 – 5 credit cards is good for your credit score.

#Now let's see the impact on credit scores based on how much average interest you pay on loans and EMIs:

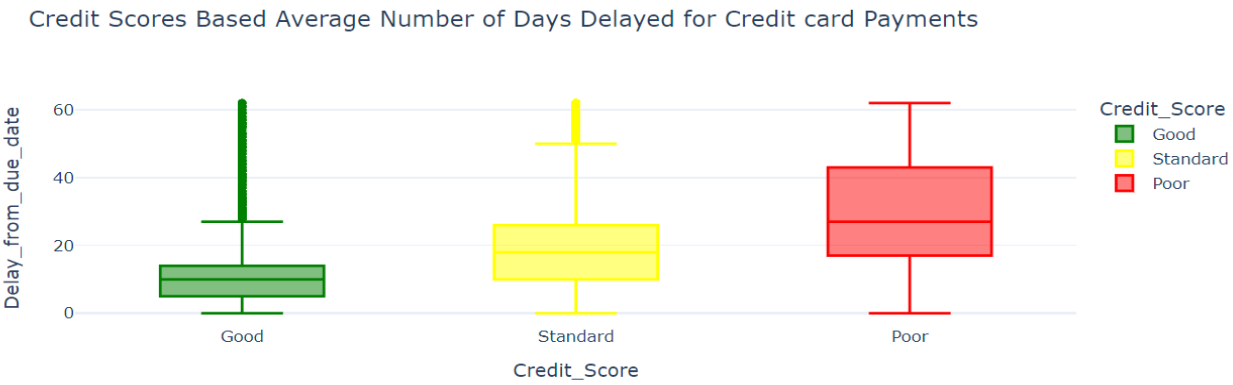
```
fig = px.box(data,
             x = "Credit_Score",
             y = "Interest_Rate",
             color = "Credit_Score",
             title ="Credit Scores Based on Interest Rates",
             color_discrete_map = {"Poor":"red",
                                   "Standard" :"yellow",
                                   "Good" : "green"})
fig.update_traces(quartilemethod = "exclusive")
fig.show()
```



#If the average interest rate is 4 – 11%, the credit score is good.Having an average interest rate of more than 15% is bad for your credit scores.

#Now let's see how many loans you can take at a time for a good credit score:

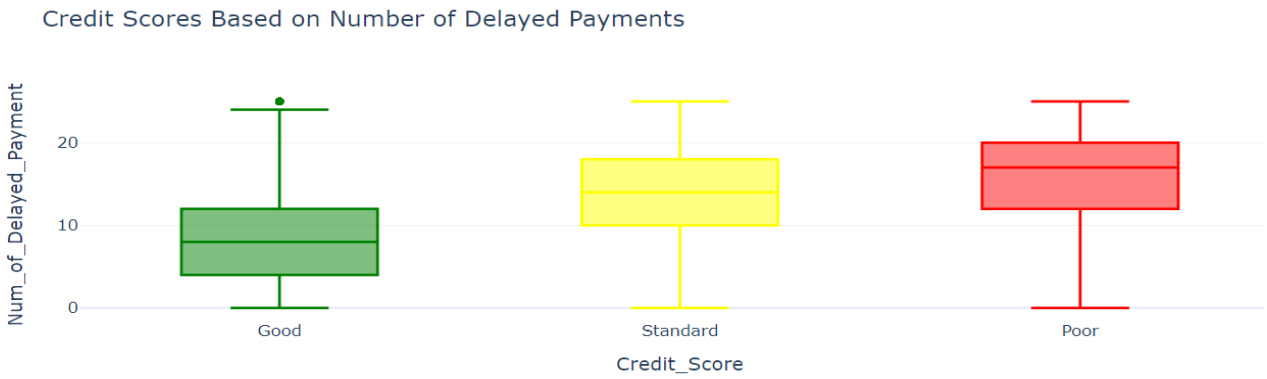
```
fig = px.box(data,
             x = "Credit_Score",
             y = "Delay_from_due_date",
             color = "Credit_Score",
             title ="Credit Scores Based Average Number of Days Delayed for Credit card Payments",
             color_discrete_map = {"Poor":"red",
                                   "Standard" :"yellow",
                                   "Good" : "green"})
fig.update_traces(quartilemethod = "exclusive")
fig.show()
```



#So you can delay your credit card payment 5 – 14 days from the due date. Delaying your payments for more than 17 days from the due date will impact your credit scores negatively.

#Now let's have a look at if frequently delaying payments will impact credit scores or not:

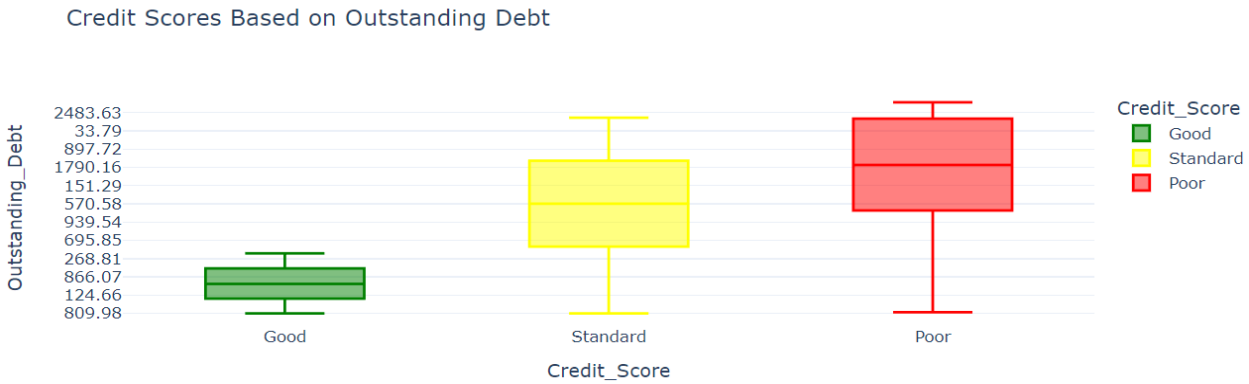
```
fig = px.box(data,
             x="Credit_Score",
             y="Num_of_Delayed_Payment",
             color="Credit_Score",
             title="Credit Scores Based on Number of Delayed Payments",
             color_discrete_map={'Poor':'red',
                                 'Standard':'yellow',
                                 'Good':'green'})
fig.update_traces(quartilemethod="exclusive")
fig.show()
```



#So, delaying 4 – 12 payments from the due date will not affect your credit scores. But delaying more than 12 payments from the due date will affect your credit scores negatively.

#Now let's see if having more debt will affect credit scores or not:

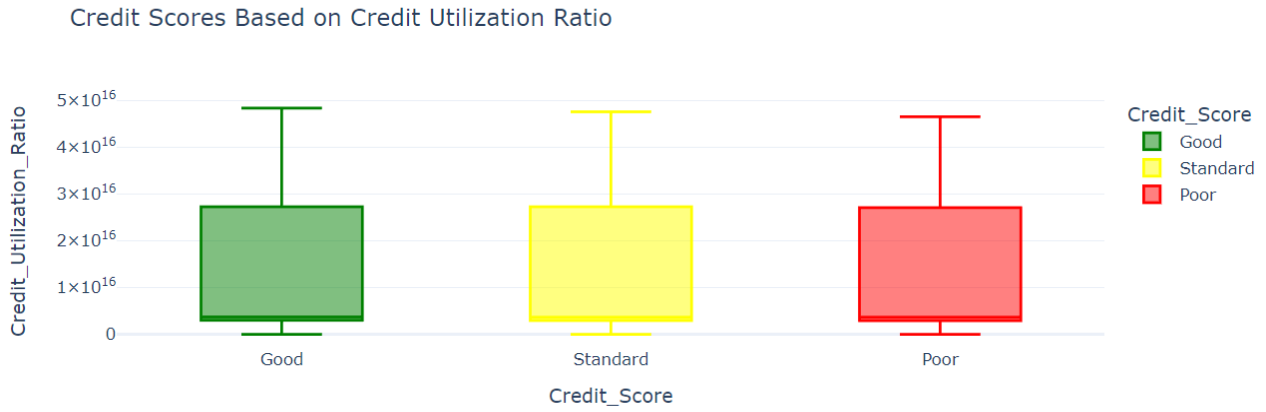
```
fig = px.box(data,
             x="Credit_Score",
             y="Outstanding_Debt",
             color="Credit_Score",
             title="Credit Scores Based on Outstanding Debt",
             color_discrete_map={'Poor':'red',
                                 'Standard':'yellow',
                                 'Good':'green'})
fig.update_traces(quartilemethod="exclusive")
fig.show()
```



#In outstanding debt of $380 – $1150 will not affect your credit scores. But always having a debt of more than $1338 will affect your credit scores negatively.

#Now let's see if having a high credit utilization ratio will affect credit scores or not:

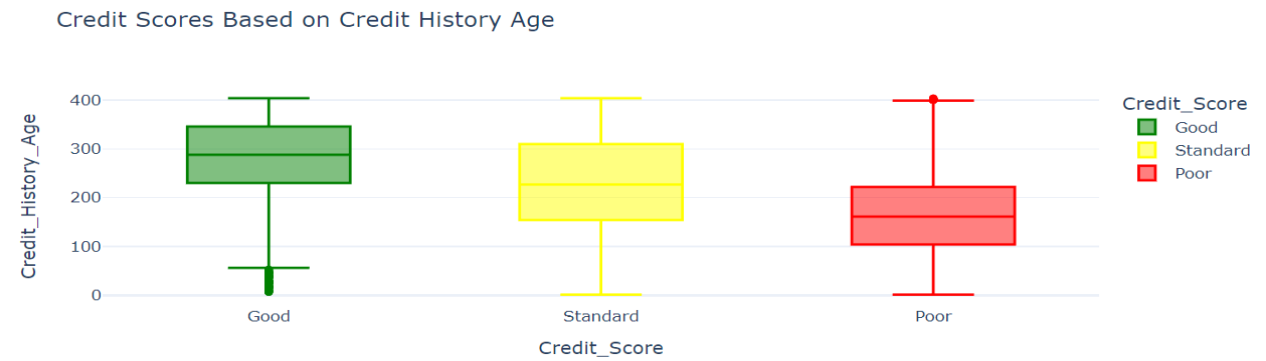
```
fig = px.box(data,
             x="Credit_Score",
             y="Credit_Utilization_Ratio",
             color="Credit_Score",
             title="Credit Scores Based on Credit Utilization Ratio",
             color_discrete_map={'Poor':'red',
                                 'Standard':'yellow',
                                 'Good':'green'})
fig.update_traces(quartilemethod="exclusive")
fig.show()
```



Credit utilization ratio means your total debt divided by your total available credit. According to the above figure, your credit utilization ratio doesn't affect your credit scores.

#Now let's see how the credit history age of a person affects credit scores:

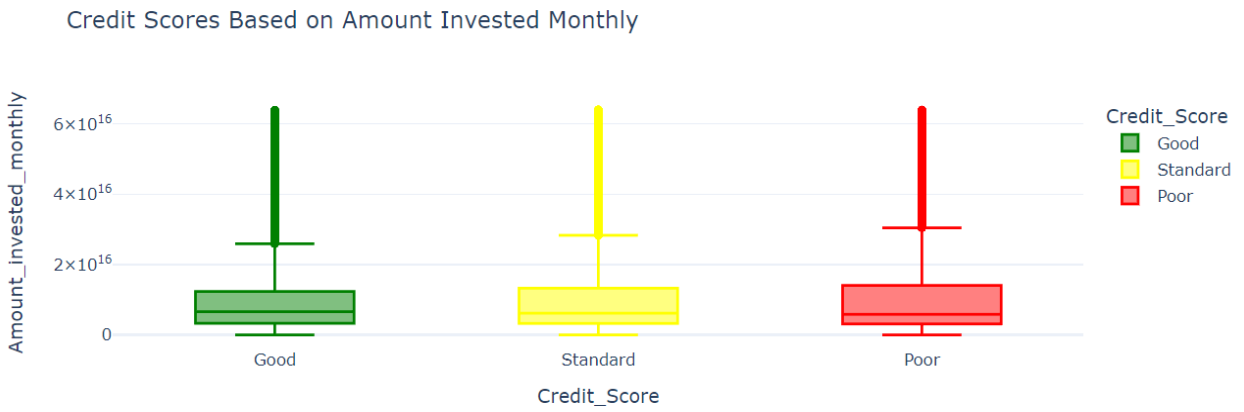
```
fig = px.box(data,
             x="Credit_Score",
             y="Credit_History_Age",
             color="Credit_Score",
             title="Credit Scores Based on Credit History Age",
             color_discrete_map={'Poor':'red',
                                 'Standard':'yellow',
                                 'Good':'green'})
fig.update_traces(quartilemethod="exclusive")
fig.show()
```



#So, having a long credit history results in better credit scores.

#Now let’s see if your monthly investments affect your credit scores or not:

```
fig = px.box(data,
             x="Credit_Score",
             y="Amount_invested_monthly",
             color="Credit_Score",
             title="Credit Scores Based on Amount Invested Monthly",
             color_discrete_map={'Poor':'red',
                                 'Standard':'yellow',
                                 'Good':'green'})
fig.update_traces(quartilemethod="exclusive")
fig.show()
```



#The amount of money you invest monthly doesn’t affect your credit scores a lot.

4.Model Development:

#As the Credit_Mix column is categorical, I will transform it into a numerical feature so that we can use it to train a Machine Learning model for the task of credit score classification:

```
data["Credit_Mix"] = data["Credit_Mix"].map({"Standart":1,
                                             "Good":2,
                                             "Bad":0})
data
```

lonthly_Inhand_Salary	Num_Bank_Accounts	...	Credit_Mix	Outstanding_Debt	Credit_Utilization_Ratio	Credit_History_Age	Payment_of_Min_Amount	Total_EMI_per_n
1.824843e+16	3.0	...	2.0	809.98	2.682262e+16	265.0	No	4.95749!
1.824843e+16	3.0	...	2.0	809.98	3.194496e+15	266.0	No	4.95749!
1.824843e+16	3.0	...	2.0	809.98	2.860935e+15	267.0	No	4.95749!
1.824843e+16	3.0	...	2.0	809.98	3.137786e+15	268.0	No	4.95749!
1.824843e+16	3.0	...	2.0	809.98	2.479735e+16	269.0	No	4.95749!
...	...	...	...	...	...	...	...	
3.359416e+15	4.0	...	2.0	502.38	3.466357e+16	378.0	No	3.51040;
3.359416e+15	4.0	...	2.0	502.38	4.056563e+16	379.0	No	3.51040;
3.359416e+15	4.0	...	2.0	502.38	4.125552e+15	380.0	No	3.51040;
3.359416e+15	4.0	...	2.0	502.38	3.363821e+15	381.0	No	3.51040;
3.359416e+15	4.0	...	2.0	502.38	3.419246e+16	382.0	No	3.51040;

#Now I will split the data into features and labels by selecting the features we found important for our model:

```
x = np.array(data[["Annual_Income", "Monthly_Inhand_Salary",
                   "Num_Bank_Accounts", "Num_Credit_Card",
                   "Interest_Rate", "Num_of_Loan",
                   "Delay_from_due_date", "Num_of_Delayed_Payment",
                   "Credit_Mix", "Outstanding_Debt",
                   "Credit_History_Age", "Monthly_Balance"]])

y = np.array(data[["Credit_Score"]])
```

#Now, let's split the data into training and test sets and proceed further by training a credit score classification model:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x, y,
                                                 test_size=0.33,
                                                 random_state=42)
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier()
model.fit(x_train,y_train)
```

#Now, let's make predictions from our model by giving inputs to our model according to the features we used to train the model:

```
print("Credit Score Prediction : ")
a = float(input("Annual Income: "))
b = float(input("Monthly Inhand Salary: "))
c = float(input("Number of Bank Accounts: "))
d = float(input("Number of Credit cards: "))
e = float(input("Interest rate: "))
f = float(input("Number of Loans: "))
g = float(input("Average number of days delayed by the person: "))
h = float(input("Number of delayed payments: "))
i = input("Credit Mix (Bad: 0, Standard: 1, Good: 3) : ")
j = float(input("Outstanding Debt: "))
k = float(input("Credit History Age: "))
l = float(input("Monthly Balance: "))

features = np.array([[a, b, c, d, e, f, g, h, i, j, k, l]])
print("Predicted Credit Score = ", model.predict(features))
```

```
Credit Score Prediction :
Annual Income:
```

5. Conclusions

This project aims to develop a model for credit score classification. The model developed using data analysis and machine learning techniques could be a valuable tool for assessing customers' credit risk. The results of the project are summarized below:

- The developed classification model has achieved high accuracy and reliability.

- The model has successfully identified significant factors affecting credit scores, ensuring accurate prediction of credit risk.

- The use of this model in evaluating credit applications could enhance the risk management processes of financial institutions and enable more effective credit decisions.

- Future work could focus on improving the model's performance through different feature selection techniques and tuning model hyperparameters.

These findings underscore the importance and impact of this study on credit score classification. Future work could involve further refinement of the model and its integration into real-world applications.

6.Recommendations

1. Further Model Refinement: Continuously refine the classification model by exploring additional features and experimenting with different machine learning algorithms. This iterative process can lead to enhanced model performance and better predictive accuracy.

2. Real-time Implementation: Consider implementing the developed model into real-time credit assessment systems used by financial institutions. This would allow for automated credit decision-making processes, resulting in faster and more efficient customer service.

3. Continuous Monitoring: Regularly monitor the model's performance and update it as needed to ensure its effectiveness in adapting to changing market dynamics and customer behaviors. This proactive approach will help maintain the model's relevance and accuracy over time.

4. Ethical Considerations: Pay close attention to ethical considerations related to data privacy, fairness, and transparency when deploying the model in real-world settings. Ensuring fairness and transparency in the credit assessment process is essential for maintaining trust and credibility with customers.

By following these recommendations, future efforts in credit score classification can build upon the foundation laid by this project and further improve the accuracy, efficiency, and fairness of credit assessment processes.

7.References

1. Peterson, T. (2022). "Understanding Credit Score Classification." Investopedia. Retrieved from: https://www.investopedia.com/understanding-credit-score-classification-5194446

2. "Credit Score Classification Case Study." Statso. Retrieved from: https://statso.io/credit-score-classification-case-study/

3. Kaggle. (2023). "Credit Score Classification Notebook." Retrieved from: https://www.kaggle.com/code/saloni1712/credit-score-classification

8.Appendices:

Appendix A: Project Code

This appendix contains the code used in the project. Python code was utilized for data analysis, model development, and obtaining results.

Appendix B: Dataset Description

This appendix provides a detailed description of the dataset used in the project. It includes information about the columns, sample records, and variables in the dataset.

Appendix C: Model Performance

This appendix presents a detailed analysis of the performance metrics evaluating the developed model. Metrics such as accuracy, precision, recall, and F1 score are examined.

Appendix D: Additional Graphs and Tables

This appendix includes additional graphs, tables, or visuals not included in the main text of the project. These supplementary materials contribute to understanding the project and visually presenting the results.

Appendix E: Used Model - Random Forest

This appendix provides a detailed description of the model used in the project. It outlines the implementation of the Random Forest algorithm, the hyperparameters used, and information about the model's performance.